The Mirage of Action-Dependent Baselines in Reinforcement Learning

George Tucker [1] **Surya Bhupatiraju** [1 2] **Shixiang Gu** [1 3 4] **Richard E. Turner** [3] **Zoubin Ghahramani** [3 5]
Sergey Levine [1 6]

Abstract

Policy gradient methods are a widely used class of model-free reinforcement learning algorithms where a state-dependent baseline is used to reduce gradient estimator variance. Several recent papers extend the baseline to depend on both the state and action and suggest that this significantly reduces variance and improves sample efficiency without introducing bias into the gradient estimates. To better understand this development, we decompose the variance of the policy gradient estimator and numerically show that learned state-action-dependent baselines do not in fact reduce variance over a state-dependent baseline in commonly tested benchmark domains. We confirm this unexpected result by reviewing the open-source code accompanying these prior papers, and show that subtle implementation decisions cause deviations from the methods presented in the papers and explain the source of the previously observed empirical gains. Furthermore, the variance decomposition highlights areas for improvement, which we demonstrate by illustrating a simple change to the typical value function parameterization that can significantly improve performance.

1. Introduction

Model-free reinforcement learning (RL) with flexible function approximators, such as neural networks (i.e., deep reinforcement learning), has shown success in goal-directed sequential decision-making problems in high dimensional state spaces (Mnih et al., 2015; Schulman et al., 2015b; Lillicrap et al., 2015; Silver et al., 2016). Policy gradient methods (Williams, 1992; Sutton et al., 2000; Kakade, 2002; Peters & Schaal, 2006; Silver et al., 2014; Schulman et al., 2015a; 2017) are a class of model-free RL algorithms that have found widespread adoption due to their stability and ease of use. Because these methods directly estimate the gradient of the expected reward RL objective, they exhibit stable convergence both in theory and practice (Sutton et al., 2000; Kakade, 2002; Schulman et al., 2015a; Gu et al., 2017b). In contrast, methods such as Q-learning lack convergence guarantees in the case of nonlinear function approximation (Sutton & Barto, 1998).

On-policy Monte-Carlo policy gradient estimates suffer from high variance, and therefore require large batch sizes to reliably estimate the gradient for stable iterative optimization (Schulman et al., 2015a). This limits their applicability to real-world problems, where sample efficiency is a critical constraint. Actor-critic methods (Sutton et al., 2000; Silver et al., 2014) and λ-weighted return estimation (Tesauro, 1995; Schulman et al., 2015b) replace the high variance Monte-Carlo return with an estimate based on the sampled return and a function approximator. This reduces variance at the expense of introducing bias from the function approximator, which can lead to instability and sensitivity to hyperparameters. In contrast, state-dependent baselines (Williams, 1992; Weaver & Tao, 2001) reduce variance without introducing bias. This is desirable because it does not compromise the stability of the original method.

Gu et al. (2017a); Grathwohl et al. (2018); Liu et al. (2018); Wu et al. (2018) present promising results extending the classic state-dependent baselines to state-action-dependent baselines. The standard explanation for the benefits of such approaches is that they achieve large reductions in variance (Grathwohl et al., 2018; Liu et al., 2018), which translates to improvements over methods that only condition the baseline on the state. This line of investigation is attractive, because by definition, baselines do not introduce bias and thus do not compromise the stability of the underlying policy gradient algorithm, but still provide improved sample efficiency. In other words, they retain the advantages of the underlying algorithms with no unintended side-effects.

In this paper, we aim to improve our understanding of state-action-dependent baselines and to identify targets for further unbiased variance reduction. Toward this goal, we present a decomposition of the variance of the policy gradient esti-

[1]Google Brain, USA [2]Work was done during the Google AI Residency. [3]University of Cambridge, UK [4]Max Planck Institute for Intelligent Systems, Germany [5]Uber AI Labs, USA [6]UC Berkeley, USA. Correspondence to: George Tucker <gjt@google.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

mator which isolates the potential variance reduction due to state-action-dependent baselines. We numerically evaluate the variance components on a synthetic linear-quadratic-Gaussian (LQG) task, where the variances are nearly analytically tractable, and on benchmark continuous control tasks and draw two conclusions: (1) on these tasks, a learned state-action-dependent baseline does not significantly reduce variance over a learned state-dependent baseline, and (2) the variance caused by using a function approximator for the value function or state-dependent baseline is much larger than the variance reduction from adding action dependence to the baseline.

To resolve the apparent contradiction arising from (1), we carefully reviewed the open-source implementations[1] accompanying Q-prop (Gu et al., 2017a), Stein control variates (Liu et al., 2018), and LAX (Grathwohl et al., 2018) and show that subtle implementation decisions cause the code to diverge from the unbiased methods presented in the papers. We explain and empirically evaluate variants of these prior methods to demonstrate that these subtle implementation details, which trade variance for bias, are in fact crucial for their empirical success. These results motivate further study of these design decisions.

The second observation (2), that function approximators poorly estimate the value function, suggests that there is room for improvement. Although many common benchmark tasks are finite horizon problems, most value function parameterizations ignore this fact. We propose a horizon-aware value function parameterization, and this improves performance compared with the state-action-dependent baseline without biasing the underlying method.

We emphasize that without the open-source code accompanying (Gu et al., 2017a; Liu et al., 2018; Grathwohl et al., 2018), this work would not be possible. Releasing the code has allowed us to present a new view on their work and to identify interesting implementation decisions for further study that the original authors may not have been aware of.

We have made our code and additional visualizations available at `https://sites.google.com/view/mirage-rl`.

2. Background

Reinforcement learning aims to learn a policy for an agent to maximize a sum of reward signals (Sutton & Barto, 1998). The agent starts at an initial state $s_0 \sim P(s_0)$. Then, the agent repeatedly samples an action a_t from a policy $\pi_\theta(a_t|s_t)$ with parameters θ, receives a reward $r_t \sim P(r_t|s_t, a_t)$, and transitions to a subsequent state s_{t+1} according to the Markovian dynamics $P(s_{t+1}|a_t, s_t)$ of the environment. This generates a trajectory of states, actions, and rewards $(s_0, a_0, r_0, s_1, a_1, \ldots)$. We abbreviate the trajectory after the initial state and action by τ.

The goal is to maximize the discounted sum of rewards along sampled trajectories

$$J(\theta) = \mathbb{E}_{s_0,a_0,\tau}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right] = \mathbb{E}_{s\sim\rho^\pi(s),a,\tau}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right],$$

where $\gamma \in [0, 1)$ is a discount parameter and $\rho^\pi(s) = \sum_{t=0}^{\infty}\gamma^t P^\pi(s_t = s)$ is the unnormalized discounted state visitation frequency.

Policy gradient methods differentiate the expected return objective with respect to the policy parameters and apply gradient-based optimization (Sutton & Barto, 1998). The policy gradient can be written as an expectation amenable to Monte Carlo estimation

$$\begin{aligned}\nabla_\theta J(\theta) &= \mathbb{E}_{s\sim\rho^\pi(s),a,\tau}\left[Q^\pi(s,a)\nabla\log\pi(a|s)\right] \\ &= \mathbb{E}_{s\sim\rho^\pi(s),a,\tau}\left[A^\pi(s,a)\nabla\log\pi(a|s)\right]\end{aligned}$$

where $Q^\pi(s,a) = \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma^t r_t|s_0 = s, a_0 = a\right]$ is the state-action value function, $V^\pi(s) = \mathbb{E}_a\left[Q^\pi(s,a)\right]$ is the value function, and $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$ is the advantage function. The equality in the last line follows from the fact that $\mathbb{E}_a\left[\nabla\log\pi(a|s)\right] = 0$ (Williams, 1992).

In practice, most policy gradient methods (including this paper) use the *undiscounted* state visitation frequencies (i.e., $\gamma = 1$ for $\rho^\pi(s)$), which produces a biased estimator for $\nabla J(\theta)$ and more closely aligns with maximizing average reward (Thomas, 2014).

We can estimate the gradient with a Monte-Carlo estimator

$$\hat{g}(s,a,\tau) = \hat{A}(s,a,\tau)\nabla\log\pi_\theta(a|s), \tag{1}$$

where $\hat{A}$ is an estimator of the advantage function up to a state-dependent constant (e.g., $\sum_t \gamma^t r_t$).

2.1. Advantage Function Estimation

Given a value function estimator, $\hat{V}(s)$, we can form a k-step advantage function estimator,

$$\hat{A}^{(k)}(s_t, a_t, \tau_{t+1}) = \sum_{i=0}^{k-1}\gamma^i r_{t+i} + \gamma^k\hat{V}(s_{t+k}) - \hat{V}(s_t),$$

where $k \in \{1, 2, ..., \infty\}$ and $\tau_{t+1} = (r_t, s_{t+1}, a_{t+1}, \ldots)$. $\hat{A}^{(\infty)}(s_t, a_t, \tau_{t+1})$ produces an unbiased gradient estimator when used in Eq. 1 regardless of the choice of $\hat{V}(s)$. However, the other estimators ($k < \infty$) produce biased estimates unless $\hat{V}(s) = V^\pi(s)$. Advantage actor critic (A2C and

[1] At the time of submission, code for (Wu et al., 2018) was not available.

A3C) methods (Mnih et al., 2016) and generalized advantage estimators (GAE) (Schulman et al., 2015b) use a single or linear combination of $\hat{A}^{(k)}$ estimators as the advantage estimator in Eq. 1. In practice, the value function estimator is never perfect, so these methods produce biased gradient estimates. As a result, the hyperparameters that control the combination of $\hat{A}^{(k)}$ must be carefully tuned to balance bias and variance (Schulman et al., 2015b), demonstrating the perils and sensitivity of biased gradient estimators. For the experiments in this paper, unless stated otherwise, we use the GAE estimator. Our focus will be on the additional bias introduced beyond that of GAE.

2.2. Baselines for Variance Reduction

The policy gradient estimator in Eq. 1 typically suffers from high variance. Control variates are a well-studied technique for reducing variance in Monte Carlo estimators without biasing the estimator (Owen, 2013). They require a correlated function whose expectation we can analytically evaluate or estimate with low variance. Because $\mathbb{E}_{a|s}\left[\nabla \log \pi(a|s)\right] = 0$, any function of the form $\phi(s)\nabla \log \pi(a|s)$ can serve as a control variate, where $\phi(s)$ is commonly referred to as a baseline (Williams, 1992). With a baseline, the policy gradient estimator becomes

$$\hat{g}(s,a,\tau) = \left(\hat{A}(s,a,\tau) - \phi(s)\right) \nabla \log \pi(a|s),$$

which does not introduce bias. Several recent methods (Gu et al., 2017a; Thomas & Brunskill, 2017; Grathwohl et al., 2018; Liu et al., 2018; Wu et al., 2018) have extended the approach to state-action-dependent baselines (i.e., $\phi(s,a)$ is a function of the state and the action). With a state-action dependent baseline $\phi(s,a)$, the policy gradient estimator is

$$\begin{aligned}\hat{g}(s,a,\tau) = &\left(\hat{A}(s,a,\tau) - \phi(s,a)\right) \nabla \log \pi(a|s) \\ &+ \nabla \mathbb{E}_{a|s}\left[\phi(s,a)\right], \end{aligned} \tag{2}$$

Now, $\nabla \mathbb{E}_{a|s}\left[\phi(s,a)\right] \neq 0$ in general, so it must be analytically evaluated or estimated with low variance for the baseline to be effective.

When the action set is discrete and not large, it is straightforward to analytically evaluate the expectation in the second term (Gu et al., 2017b; Gruslys et al., 2017). In the continuous action case, Gu et al. (2017a) set $\phi(s,a)$ to be the first order Taylor expansion of a learned advantage function approximator. Because $\phi(s,a)$ is linear in a, the expectation can be analytically computed. Gu et al. (2017b); Liu et al. (2018); Grathwohl et al. (2018) set $\phi(s,a)$ to be a learned function approximator and leverage the reparameterization trick to estimate $\nabla \mathbb{E}_{a|s}\left[\phi(s,a)\right]$ with low variance when π is reparameterizable (Kingma & Welling, 2013; Rezende et al., 2014).

3. Policy Gradient Variance Decomposition

Now, we analyze the variance of the policy gradient estimator with a state-action dependent baseline (Eq. 2). This is an unbiased estimator of $\mathbb{E}_{s,a,\tau}\left[\hat{A}(s,a,\tau)\nabla \log \pi(a|s)\right]$ for any choice of ϕ. For theoretical analysis, we assume that we can analytically evaluate the expectation over a in the second term because it only depends on ϕ and π, which we can evaluate multiple times without querying the environment.

The variance of the policy gradient estimator in Eq. 2, $\Sigma := \mathrm{Var}_{s,a,\tau}(\hat{g})$, can be decomposed using the law of total variance as

$$\begin{aligned}\Sigma = &\mathbb{E}_s\left[\mathrm{Var}_{a,\tau|s}\left(\left(\hat{A}(s,a,\tau) - \phi(s,a)\right)\nabla \log \pi(a|s)\right)\right] \\ &+ \mathrm{Var}_s\, \mathbb{E}_{a,\tau|s}\left[\hat{A}(s,a,\tau)\nabla \log \pi(a|s)\right],\end{aligned}$$

where the simplification of the second term is because the baseline does not introduce bias. We can further decompose the first term,

$$\begin{aligned}&\mathbb{E}_s\left[\mathrm{Var}_{a,\tau|s}\left(\left(\hat{A}(s,a,\tau) - \phi(s,a)\right)\nabla \log \pi(a|s)\right)\right] \\ &= \mathbb{E}_{s,a}\left[\mathrm{Var}_{\tau|s,a}\left(\hat{A}(s,a,\tau)\nabla \log \pi(a|s)\right)\right] \\ &\quad + \mathbb{E}_s\left[\mathrm{Var}_{a|s}\left(\left(\hat{A}(s,a) - \phi(s,a)\right)\nabla \log \pi(a|s)\right)\right],\end{aligned}$$

where $\hat{A}(s,a) = \mathbb{E}_{\tau|s,a}\left[\hat{A}(s,a,\tau)\right]$. Putting the terms together, we arrive at the following:

$$\begin{aligned}\Sigma = &\underbrace{\mathbb{E}_{s,a}\left[\mathrm{Var}_{\tau|s,a}\left(\hat{A}(s,a,\tau)\nabla \log \pi(a|s)\right)\right]}_{\Sigma_\tau} \\ &+ \underbrace{\mathbb{E}_s\left[\mathrm{Var}_{a|s}\left(\left(\hat{A}(s,a) - \phi(s,a)\right)\nabla \log \pi(a|s)\right)\right]}_{\Sigma_a} \\ &+ \underbrace{\mathrm{Var}_s\left(\mathbb{E}_{a|s}\left[\hat{A}(s,a)\nabla \log \pi(a|s)\right]\right)}_{\Sigma_s}.\end{aligned} \tag{3}$$

Notably, only Σ_a involves ϕ, and it is clear that the variance minimizing choice of $\phi(s,a)$ is $\hat{A}(s,a)$. For example, if $\hat{A}(s,a,\tau) = \sum_t \gamma^t r_t$, the discounted return, then the optimal choice of $\phi(s,a)$ is $\hat{A}(s,a) = \mathbb{E}_{\tau|s,a}\left[\sum_t \gamma^t r_t\right] = Q^\pi(s,a)$, the state-action value function.

The variance in the on-policy gradient estimate arises from the fact that we only collect data from a limited number of states s, that we only take a single action a in each state, and that we only rollout a single path from there on τ. Intuitively, Σ_τ describes the variance due to sampling a single τ, Σ_a describes the variance due to sampling a single a, and lastly Σ_s describes the variance coming from visiting a limited number of states. The magnitudes of these terms depends on task specific parameters and the policy.

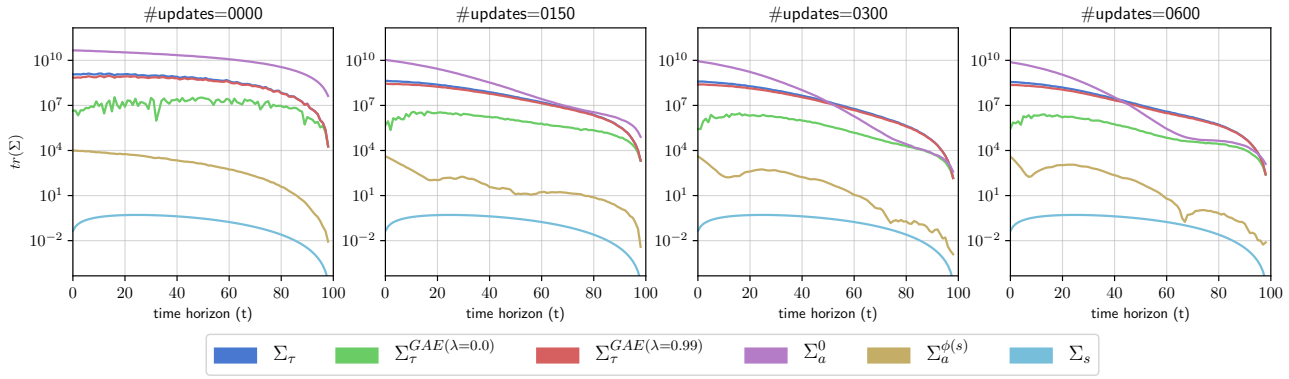

Figure 1. Evaluating the variance terms (Eq. 3) of the policy gradient estimator on a 2D point mass task (an LQG system) with finite horizon $T = 100$. The total variance of the gradient estimator covariance is plotted against time in the task (t). Each plot from left to right corresponds to a different stage in learning (see Appendix 9 for policy visualizations), and its title indicates the number of policy updates completed. Σ_a^0 and $\Sigma_a^{\phi(s)}$ correspond to the Σ_a term without a baseline and using the value function as a state-dependent baseline, respectively. Importantly, an optimal state-action-dependent baseline reduces Σ_a to 0, so $\Sigma_a^{\phi(s)}$ upper bounds the variance reduction possible from using a state-action-dependent baseline over a state-dependent baseline. In this task, $\Sigma_a^{\phi(s)}$ is much smaller than Σ_τ, so the reduction in overall variance from using a state-action-dependent baseline would be minimal. $\Sigma_\tau^{GAE(\lambda)}$ indicates the Σ_τ term with GAE-based return estimates. We include animated GIF visualizations of the variance terms and policy as learning progresses in the Supplementary Materials.

The relative magnitudes of the variance terms will determine the effectiveness of the optimal state-action-dependent baseline. In particular, denoting the value of the second term when using a state-dependent baseline by $\Sigma_a^{\phi(s)}$, the variance of the policy gradient estimator with a state-dependent baseline is $\Sigma_a^{\phi(s)} + \Sigma_\tau + \Sigma_s$. When $\phi(s, a)$ is optimal, Σ_a vanishes, so the variance is $\Sigma_\tau + \Sigma_s$. Thus, an optimal state-action-dependent baseline will be beneficial when $\Sigma_a^{\phi(s)}$ is large relative to $\Sigma_\tau + \Sigma_s$. We expect this to be the case when single actions have a large effect on the overall discounted return (e.g., in a Cliffworld domain, where a single action could cause the agent to fall of the cliff and suffer a large negative reward). Practical implementations of a state-action-dependent baseline require learning $\phi(s, a)$, which will further restrict the potential benefits.

3.1. Variance in LQG Systems

Linear-quadratic-Gaussian (LQG) systems (Stengel, 1986) are a family of widely studied continuous control problems with closed-form expressions for optimal controls, quadratic value functions, and Gaussian state marginals. We first analyze the variance decomposition in an LQG system because it allows nearly analytic measurement of the variance terms in Eq. 3 (See Appendix 9 for measurement details).

Figure 1 plots the variance terms for a simple 2D point mass task using discounted returns as the choice of $\hat{A}(s, a, \tau)$ (See Appendix 9 for task details). As expected, without a baseline ($\phi = 0$), the variance of Σ_a^0 is much larger than Σ_τ and Σ_s. Further, using the value function as a state-dependent baseline ($\phi(s) = V^\pi(s)$), results in a large variance reduction (compare the lines for $\Sigma_a^{\phi(s)}$ and Σ_a^0 in Figure 1). An optimal state-action-dependent baseline would reduce $\Sigma_a^{\phi(s)}$ to 0, however, for this task, such a baseline would not significantly reduce the total variance because Σ_τ is already large relative to $\Sigma_a^{\phi(s)}$ (Figure 1).

We also plot the effect of using GAE[2] (Schulman et al., 2015b) on Σ_τ for $\lambda = \{0, 0.99\}$. Baselines and GAE reduce different components of the gradient variances, and this figure compares their effects throughout the learning process.

3.2. Empirical Variance Measurements

We estimate the magnitude of the three terms for benchmark continuous action tasks as training proceeds. Once we decide on the form of $\phi(s, a)$, approximating ϕ is a learning problem in itself. To understand the approximation error, we evaluate the situation where we have access to an oracle $\phi(s, a)$ and when we learn a function approximator for $\phi(s, a)$. Estimating the terms in Eq. 3 is nontrivial because the expectations and variances are not available in closed form. We construct unbiased estimators of the variance terms and repeatedly draw samples to drive down the measurement error (see Appendix 10 for details). We train a

[2]For the LQG system, we use the oracle value function to compute the GAE estimator. In the rest of the experiments, GAE is computed using a learned value function.

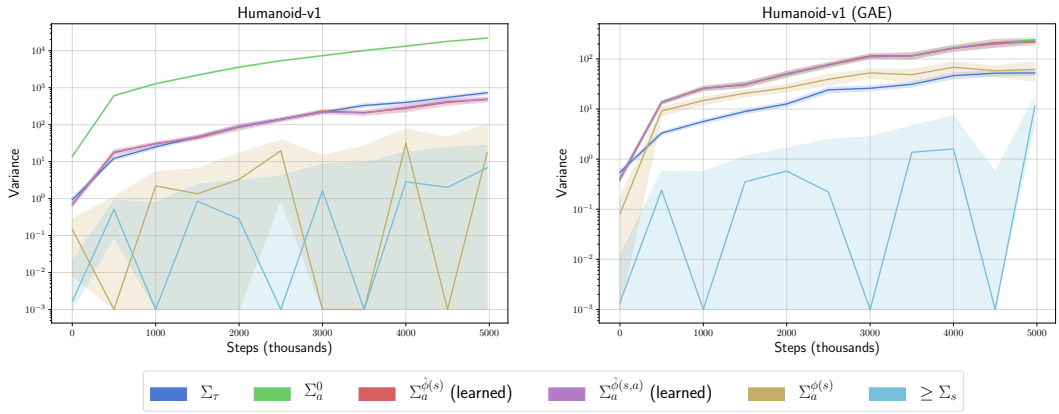

Figure 2. Evaluating the variance terms (Eq. 3) of the gradient estimator when $\hat{A}(s,a,\tau)$ is the discounted return (left) and GAE (right) with various baselines on Humanoid (See Appendix Figure 9 for results on HalfCheetah). The x-axis denotes the number of environment steps used for training. The policy is trained with TRPO. We set $\phi(s) = \mathbb{E}_{a|s}\left[\hat{A}(s,a)\right]$ and $\phi(s,a) = \hat{A}(s,a)$. The "learned" label in the legend indicates that a function approximator to ϕ was used instead of directly using ϕ. Note that when using $\phi(s,a) = \hat{A}(s,a)$, $\Sigma_a^{\phi(s,a)}$ is 0, so is not plotted. Since Σ_s is small, we plot an upper bound on Σ_s. The upper and lower bands indicate two standard errors of the mean. In the left plot, lines for $\Sigma_a^{\hat{\phi}(s)}$ and $\Sigma_a^{\hat{\phi}(s,a)}$ overlap and in the right plot, lines for Σ_a^0, $\Sigma_a^{\hat{\phi}(s)}$, and $\Sigma_a^{\hat{\phi}(s,a)}$ overlap.

policy using TRPO[3] (Schulman et al., 2015a) and as training proceeds, we plot each of the individual terms Σ_τ, Σ_a, and Σ_s of the gradient estimator variance for Humanoid in Figure 2 and for HalfCheetah in Appendix Figure 9. Additionally, we repeat the experiment with the horizon-aware value functions (described in Section 5) in Appendix Figures 10 and 11.

We plot the variance decomposition for two choices of $\hat{A}(s,a,\tau)$: the discounted return, $\sum_t \gamma^t r_t$, and GAE (Schulman et al., 2015b). In both cases, we set $\phi(s) = \mathbb{E}_{a|s}\left[\hat{A}(s,a)\right]$ and $\phi(s,a) = \hat{A}(s,a)$ (the optimal state-action-dependent baseline). When using the discounted return, we found that Σ_τ dominates $\Sigma_a^{\phi(s)}$, suggesting that even an optimal state-action-dependent baseline (which would reduce Σ_a to 0) would not improve over a state-dependent baseline (Figure 2). In contrast, with GAE, Σ_τ is reduced and now the optimal state-action-dependent baseline would reduce the overall variance compared to a state-dependent baseline. However, when we used function approximators to ϕ, we found that the state-dependent and state-action-dependent function approximators produced similar variance and much higher variance than when using an oracle ϕ (Figure 2). This suggests that, in practice, we would not see improved learning performance using a state-action-dependent baseline over a state-dependent baseline on these tasks. We confirm this in later experiments in Sections 4 and 5.

[3]The relative magnitudes of the variance terms depend on the task, policy, and network structures. For evaluation, we use a well-tuned implementation of TRPO (Appendix 8.4).

Furthermore, we see that closing the function approximation gap of $V(s)$ and $\phi(s)$ would produce much larger reductions in variance than from using the optimal state-action-dependent baseline over the state-dependent baseline. This suggests that improved function approximation of both $V(s)$ and $\phi(s)$ should be a priority. Finally, Σ_s is relatively small in both cases, suggesting that focusing on reducing variance from the first two terms of Eq. 3, Σ_τ and Σ_a, will be more fruitful.

4. Unveiling the Mirage

In the previous section, we decomposed the policy gradient variance into several sources, and we found that in practice, the source of variance reduced by the state-action-dependent baseline is not reduced when a function approximator for ϕ is used. However, this appears to be a paradox: if the state-action-dependent baseline does not reduce variance, how are prior methods that propose state-action-dependent baselines able to report significant improvements in learning performance? We analyze implementations accompanying these works, and show that they actually introduce bias into the policy gradient due to subtle implementation decisions[4].

[4]The implementation of the state-action-dependent baselines for continuous control in (Grathwohl et al., 2018) suffered from two critical issues (see Appendix 8.3 for details), so it was challenging to determine the source of their observed performance. After correcting these issues in their implementation, we do not observe an improvement over a state-dependent baseline, as shown in Appendix Figure 13. We emphasize that these observations are restricted to the continuous control experiments as the rest of the experiments in that paper use a separate codebase that is unaffected.

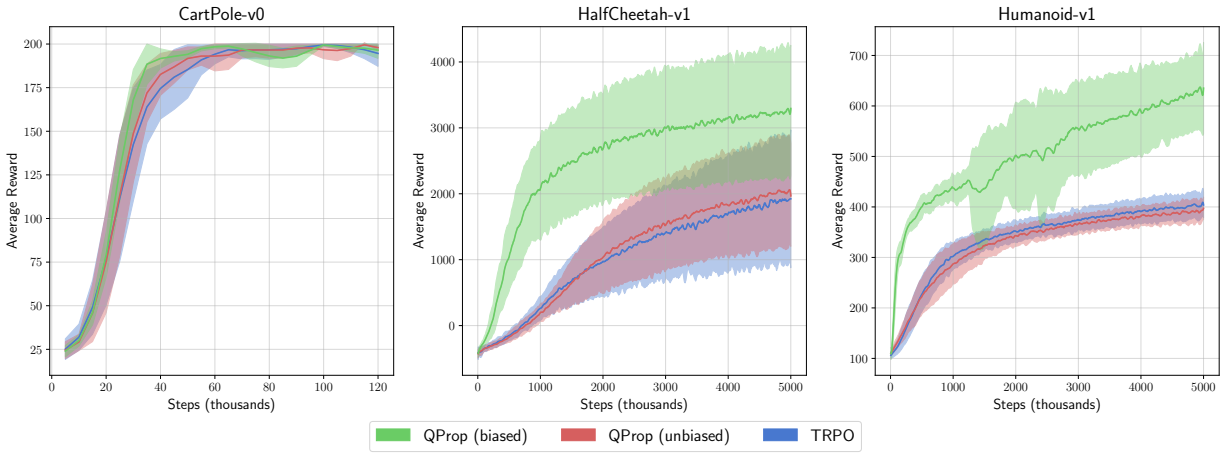

Figure 3. Evaluation of Q-Prop, an unbiased version of Q-Prop that applies the normalization to all terms, and TRPO (implementations based on the code accompanying Gu et al. (2017a)). We plot mean episode reward with standard deviation intervals capped at the minimum and maximum across 10 random seeds. The batch size across all experiments was 5000. On the continuous control tasks (HalfCheetah and Humanoid), we found that that the unbiased Q-Prop performs similarly to TRPO, while the (biased) Q-Prop outperforms both. On the discrete task (CartPole), we found almost no difference between the three algorithms.

We find that these methods are effective not because of unbiased variance reduction, but instead because they introduce bias for variance reduction.

4.1. Advantage Normalization

Although Q-Prop and IPG (Gu et al., 2017b) (when $\nu = 0$) claim to be unbiased, the implementations of Q-Prop and IPG apply an adaptive normalization to only some of the estimator terms, which introduces a bias. Practical implementations of policy gradient methods (Mnih & Gregor, 2014; Schulman et al., 2015b; Duan et al., 2016) often normalize the advantage estimate $\hat{A}$, also commonly referred to as the *learning signal*, to unit variance with batch statistics. This effectively serves as an adaptive learning rate heuristic that bounds the gradient variance.

The implementations of Q-Prop and IPG normalize the learning signal $\hat{A}(s,a,\tau) - \phi(s,a)$, but not the bias correction term $\nabla \mathbb{E}_a [\phi(s,a)]$. Explicitly, the estimator with such a normalization is,

$$\hat{g}(s,a,\tau) = \frac{1}{\hat{\sigma}} \left(\hat{A}(s,a,\tau) - \phi(s,a) - \hat{\mu} \right) \nabla \log \pi(a|s) + \nabla \mathbb{E}_{a|s} [\phi(s,a)] ,$$

where $\hat{\mu}$ and $\hat{\sigma}$ are batch-based estimates of the mean and standard deviation of $\hat{A}(s,a,\tau) - \phi(s,a)$. This deviates from the method presented in the paper and introduces bias. In fact, IPG (Gu et al., 2017b) analyzes the bias in the implied objective that would be introduced when the first term has a different weight from the bias correction term, proposing such a weight as a means to trade off bias and variance. We analyze the bias and variance of the gradient estimator in Appendix 11. However, the weight actually used in the implementation is off by the factor $\hat{\sigma}$, and never one (which corresponds to the unbiased case). This introduces an adaptive bias-variance trade-off that constrains the learning signal variance to 1 by automatically adding bias if necessary.

In Figure 3, we compare the implementation of Q-Prop from (Gu et al., 2017a), an unbiased implementation of Q-Prop that applies the normalization to all terms, and TRPO. We found that the adaptive bias-variance trade-off induced by the asymmetric normalization is crucial for the gains observed in (Gu et al., 2017a). If implemented as unbiased, it does not outperform TRPO.

4.2. Poorly Fit Value Functions

In contrast to our results, Liu et al. (2018) report that state-action-dependent baselines significantly reduce variance over state-dependent baselines on continuous action benchmark tasks (in some cases by six orders of magnitude). We find that this conclusion was caused by a poorly fit value function.

The GAE advantage estimator has mean zero when $\hat{V}(s) = V^\pi(s)$, which suggests that a state-dependent baseline is unnecessary if $\hat{V}(s) \approx V^\pi(s)$. As a result, a state-dependent baseline is typically omitted when the GAE advantage estimator is used. This is the case in (Liu et al., 2018). However, when $\hat{V}(s)$ poorly approximates $V^\pi(s)$, the GAE advantage estimator has nonzero mean, and a state-dependent baseline can reduce variance. We show that is the case by taking the open-source code accompanying (Liu et al., 2018), and implementing a state-dependent baseline. It achieves comparable variance reduction to the state-action-dependent

baseline (Appendix Figure 12).

This situation can occur when the value function approximator is not trained sufficiently (e.g., if a small number of SGD steps are used to train $\hat{V}(s)$). Then, it can appear that adding a state-action-dependent baseline reduces variance where a state-dependent baseline would have the same effect.

4.3. Sample-Reuse in Baseline Fitting

Recent work on state-action-dependent baselines fits the baselines using on-policy samples (Liu et al., 2018; Grathwohl et al., 2018) either by regressing to the Monte Carlo return or minimizing an approximation to the variance of the gradient estimator. This must be carefully implemented to avoid bias. Specifically, fitting the baseline to the current batch of data and then using the updated baseline to form the estimator results in a biased gradient (Jie & Abbeel, 2010).

Although this can reduce the variance of the gradient estimator, it is challenging to analyze the bias introduced. The bias is controlled by the implicit or explicit regularization (e.g., early stopping, size of the network, etc.) of the function approximator used to fit ϕ. A powerful enough function approximator can trivially overfit the current batch of data and reduce the learning signal to 0. This is especially important when flexible neural networks are used as the function approximators.

Liu et al. (2018) fit the baseline using the current batch before computing the policy gradient estimator. Using the open-source code accompanying (Liu et al., 2018), we evaluate several variants: an unbiased version that fits the state-action-dependent baseline after computing the policy step, an unbiased version that fits a state-dependent baseline after computing the policy step, and a version that estimates $\nabla E_{a|s}\left[\phi(s,a)\right]$ with an extra sample of $a \sim \pi(a|s)$ instead of importance weighting samples from the current batch. Our results are summarized in Appendix Figure 8. Notably, we found that using an extra sample, which should reduce variance by avoiding importance sampling, decreases performance because the baseline is overfit to the current batch. The performance of the unbiased state-dependent baseline matched the performance of the unbiased state-action-dependent baseline. On Humanoid, the biased method implemented in (Liu et al., 2018) performs best. However, on HalfCheetah, the biased methods suffer from instability.

5. Horizon-Aware Value Functions

The empirical variance decomposition illustrated in Figure 2 and Appendix Figure 9 reveals deficiencies in the commonly used value function approximator, and as we showed in Section 4.2, a poor value approximator can produce misleading results. To fix one deficiency with the value function approximator, we propose a new horizon-aware parameterization of the value function. As with the state-action-dependent baselines, such a modification is appealing because it does not introduce bias into the underlying method.

The standard continuous control benchmarks use a fixed time horizon (Duan et al., 2016; Brockman et al., 2016), yet most value function parameterizations are stationary, as though the task had infinite horizon. Near the end of an episode, the expected return will necessarily be small because there are few remaining steps to accumulate reward. To remedy this, our value function approximator outputs two values: $\hat{r}(s)$ and $\hat{V}'(s)$ and then we combine them with the discounted time left to form a value function estimate

$$\hat{V}(s_t) = \left(\sum_{i=t}^{T} \gamma^{i-t}\right) \hat{r}(s_t) + \hat{V}'(s_t),$$

where T is the maximum length of the episode. Conceptually, we can think of $\hat{r}(s)$ as predicting the average reward over future states and $\hat{V}'(s)$ as a state-dependent offset. $\hat{r}(s)$ is a rate of return, so we multiply it be the remaining discounted time in the episode.

Including time as an input to the value function can also resolve this issue (e.g., (Duan et al., 2016; Pardo et al., 2017)). We compare our horizon-aware parameterization against including time as an input to the value function and find that the horizon-aware value function performs favorably (Appendix Figures 6 and 7).

In Figure 4, we compare TRPO with a horizon-aware value function against TRPO, TRPO with a state-dependent baseline, and TRPO with a state-action-dependent baseline. Across environments, the horizon-aware value function outperforms the other methods. By prioritizing the largest variance components for reduction, we can realize practical performance improvements without introducing bias.

6. Related Work

Baselines (Williams, 1992; Weaver & Tao, 2001) in RL fall under the umbrella of control variates, a general technique for reducing variance in Monte Carlo estimators without biasing the estimator (Owen, 2013). Weaver & Tao (2001) analyzes the optimal state-dependent baseline, and in this work, we extend the analysis to state-action-dependent baselines in addition to analyzing the variance of the GAE estimator (Tesauro, 1995; Schulman et al., 2015a).

Dudík et al. (2011) introduced the community to doubly-robust estimators, a specific form of control variate, for off-policy evaluation in bandit problems. The state-action-dependent baselines (Gu et al., 2017a; Wu et al., 2018; Liu et al., 2018; Grathwohl et al., 2018; Gruslys et al., 2017) can be seen as the natural extension of the doubly robust estimator to the policy gradient setting. In fact, for the discrete action case, the policy gradient estimator with the

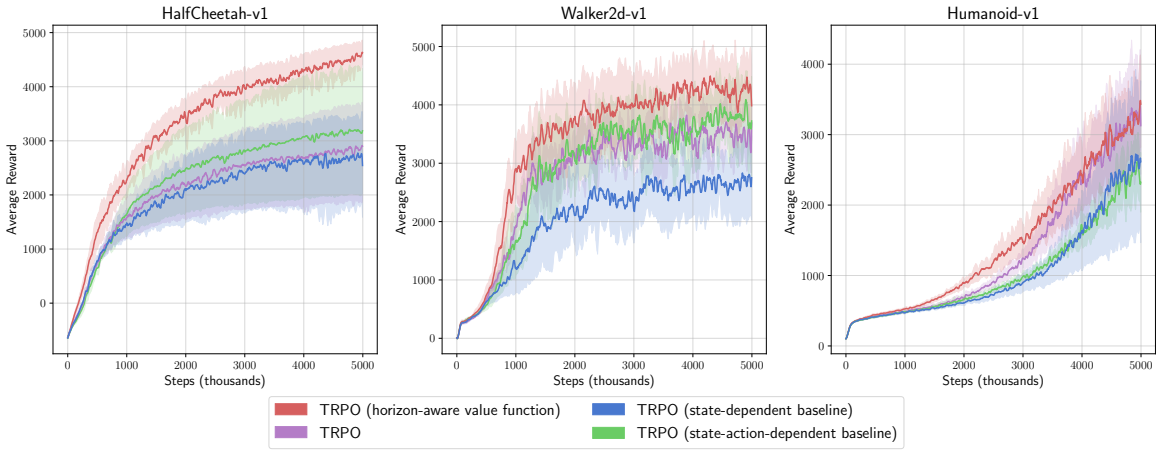

Figure 4. Evaluating the horizon-aware value function, TRPO with a state-dependent baseline, TRPO state-action-dependent baseline, and TRPO. We plot mean episode reward and standard deviation intervals capped at the minimum and maximum across 5 random seeds. The batch size across all experiments was 5000.

state-action-dependent baseline can be seen as the gradient of a doubly robust estimator.

Prior work has explored model-based (Sutton, 1990; Heess et al., 2015; Gu et al., 2016) and off-policy critic-based gradient estimators (Lillicrap et al., 2015). In off-policy evaluation, practitioners have long realized that constraining the estimator to be unbiased is too limiting. Instead, recent methods mix unbiased doubly-robust estimators with biased model-based estimates and minimize the mean squared error (MSE) of the combined estimator (Thomas & Brunskill, 2016; Wang et al., 2016a). In this direction, several recent methods have successfully mixed high-variance, unbiased on-policy gradient estimates directly with low-variance, biased off-policy or model-based gradient estimates to improve performance (O'Donoghue et al., 2016; Wang et al., 2016b; Gu et al., 2017b). It would be interesting to see if the ideas from off-policy evaluation could be further adapted to the policy gradient setting.

7. Discussion

State-action-dependent baselines promise variance reduction without introducing bias. In this work, we clarify the practical effect of state-action-dependent baselines in common continuous control benchmark tasks. Although an optimal state-action-dependent baseline is guaranteed not to increase variance and has the potential to reduce variance, in practice, currently used function approximators for the state-action-dependent baselines are unable to achieve significant variance reduction. Furthermore, we found that much larger gains could be achieved by instead improving the accuracy of the value function or the state-dependent baseline function approximators.

With these insights, we re-examined previous work on state-action-dependent baselines and identified a number of pitfalls. We were also able to correctly attribute the previously observed results to implementation decisions that introduce bias in exchange for variance reduction. We intend to further explore the trade-off between bias and variance in future work.

Motivated by the gap between the value function approximator and the true value function, we propose a novel modification of the value function parameterization that makes it aware of the finite time horizon. This gave consistent improvements over TRPO, whereas the unbiased state-action-dependent baseline did not outperform TRPO.

Finally, we note that the relative contributions of each of the terms to the policy gradient variance are problem specific. A learned state-action-dependent baseline will be beneficial when $\Sigma_a^{\hat{\phi}(s)}$ is large relative to $\Sigma_\tau + \Sigma_s$. In this paper, we focused on continuous control benchmarks where we found this not to be the case. We speculate that in environments where single actions have a strong influence on the discounted return (and hence $\mathrm{Var}_a(A(s,a))$ is large), Σ_a may be large. For example, in a discrete task with a critical decision point such as a Cliffworld domain, where a single action could cause the agent to fall of the cliff and suffer a large negative reward. Future work will investigate the variance decomposition in additional domains.

Acknowledgments

We thank Jascha Sohl-Dickstein, Luke Metz, Gerry Che, Yuchen Lu, and Cathy Wu for helpful discussions. We thank Hao Liu and Qiang Liu for assisting our understanding of their code. SG acknowledges support from a Cambridge-Tübingen PhD Fellowship. RET acknowledges support from Google and EPSRC grants EP/M0269571 and

EP/L000776/1. ZG acknowledges support from EPSRC grant EP/J012300/1.

References

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pp. 1329–1338, 2016.

Dudík, M., Langford, J., and Li, L. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, 2011.

Grathwohl, W., Choi, D., Wu, Y., Roeder, G., and Duvenaud, D. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *International Conference on Learning Representations (ICLR)*, 2018.

Gruslys, A., Azar, M. G., Bellemare, M. G., and Munos, R. The reactor: A sample-efficient actor-critic architecture. *arXiv preprint arXiv:1704.04651*, 2017.

Gu, S., Lillicrap, T., Sutskever, I., and Levine, S. Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning*, pp. 2829–2838, 2016.

Gu, S., Lillicrap, T., Ghahramani, Z., Turner, R. E., and Levine, S. Q-prop: Sample-efficient policy gradient with an off-policy critic. *International Conference on Learning Representations (ICLR)*, 2017a.

Gu, S., Lillicrap, T., Turner, R. E., Ghahramani, Z., Schölkopf, B., and Levine, S. Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 3849–3858, 2017b.

Heess, N., Wayne, G., Silver, D., Lillicrap, T., Erez, T., and Tassa, Y. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, pp. 2944–2952, 2015.

Jie, T. and Abbeel, P. On a connection between importance sampling and the likelihood ratio policy gradient. In *Advances in Neural Information Processing Systems*, pp. 1000–1008, 2010.

Kakade, S. M. A natural policy gradient. In *Advances in Neural Information Processing Systems*, pp. 1531–1538, 2002.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Liu, D. C. and Nocedal, J. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

Liu, H., Feng, Y., Mao, Y., Zhou, D., Peng, J., and Liu, Q. Action-dependent control variates for policy optimization via stein identity. *International Conference on Learning Representations (ICLR)*, 2018.

Mnih, A. and Gregor, K. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529, 2015.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pp. 1928–1937, 2016.

O'Donoghue, B., Munos, R., Kavukcuoglu, K., and Mnih, V. Pgq: Combining policy gradient and q-learning. *arXiv preprint arXiv:1611.01626*, 2016.

Owen, A. B. Monte carlo theory, methods and examples. *Monte Carlo Theory, Methods and Examples. Art Owen*, 2013.

Pardo, F., Tavakoli, A., Levdik, V., and Kormushev, P. Time limits in reinforcement learning. *arXiv preprint arXiv:1712.00378*, 2017.

Peters, J. and Schaal, S. Policy gradient methods for robotics. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 2219–2225. IEEE, 2006.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015a.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *ICML*, 2014.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Stengel, R. F. *Optimal control and estimation*. Courier Corporation, 1986.

Sutton, R. S. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine Learning Proceedings 1990*, pp. 216–224. Elsevier, 1990.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*, volume 1. MIT Press Cambridge, 1998.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1063, 2000.

Tesauro, G. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Thomas, P. Bias in natural actor-critic algorithms. In *International Conference on Machine Learning*, pp. 441–448, 2014.

Thomas, P. and Brunskill, E. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 2139–2148, 2016.

Thomas, P. S. and Brunskill, E. Policy gradient methods for reinforcement learning with function approximation and action-dependent baselines. *arXiv preprint arXiv:1706.06643*, 2017.

Wang, Y.-X., Agarwal, A., and Dudik, M. Optimal and adaptive off-policy evaluation in contextual bandits. *arXiv preprint arXiv:1612.01205*, 2016a.

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., and de Freitas, N. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016b.

Weaver, L. and Tao, N. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pp. 538–545. Morgan Kaufmann Publishers Inc., 2001.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pp. 5–32. Springer, 1992.

Wu, C., Rajeswaran, A., Duan, Y., Kumar, V., Bayen, A. M., Kakade, S., Mordatch, I., and Abbeel, P. Variance reduction for policy gradient with action-dependent factorized baselines. *International Conference on Learning Representations (ICLR)*, 2018.